INTEGRATED NAVIGATION AND GUIDANCE FOR PRECISION LANDING AT MARS

P. Daniel BURKHART
Jet Propulsion Laboratory
4800 Oak Grove Drive M/S 301-125L, Pasadena, CA 91109
(818) 393-7959 (Phone), (818) 393-6388 (Fax), dan.burkhart@jpl.nasa.gov

Robert H. BISHOP
The Center for Space Research
The University of Texas at Austin, Austin, TX 78712
(512) 471-7994 (Phone), (512) 471-3570 (Fax), bishop@csr.utexas.edu

Timothy P. CRAIN
NASA Johnson Space Center
EG5 Advanced Mission Design, Houston, TX 77058
(281) 244-5077 (Phone), (281) 483-1329 (Fax), tim.crain1@jsc.nasa.gov

ABSTRACT *- The primary requirement of next-generation entry descent and landing (EDL) systems is to reduce the size of the landing zone for a Martian lander from hundreds of kilometers to under ten kilometers. The NASA effort, led by the Jet Propulsion Laboratory, relies on several techniques never before utilized at Mars which require accurate knowledge of the position, velocity, and attitude of the spacecraft in real time. This focus of this paper is the interaction of the aided-navigation system with the rest of the flight system. Results from simulated EDL scenarios will be presented.*

KEYWORDS: Kalman filter, EDL, real-time, simulation

INTRODUCTION

The primary requirement of the next generation entry descent and landing (EDL) systems is to reduce the size of the landing zone for a Martian lander from the currently acceptable uncertainty of hundreds of kilometers to under ten kilometers. The NASA effort, led by the Jet Propulsion Laboratory, relies on several techniques never before utilized at Mars, namely, a guided entry and an altitude-velocity dependent parachute deploy logic. The proposed Mars Smart Lander will be the first demonstration of a real-time onboard precision navigation and guidance capability at Mars. In addition to reducing the landing site footprint, the Mars Smart Lander will demonstrate terminal phase hazard avoidance. The concept is to use onboard sensors to generate a terrain map of the proposed landing site and surrounding area. The system will process the terrain data and determine a safe zone within the scan area. If the projected landing site is deemed hazardous, a new landing site is selected and the spacecraft automatically retargets the new site. Together, these new technologies will allow a lander to target areas currently deemed too hazardous but are of significant interest to the science community.

An advanced technology development program has been initiated to develop the new technologies required to achieve the required landing accuracies for the next generation Martian landers. The

guided entry and hazard avoidance systems require accurate knowledge of the position, velocity, and attitude (i.e., the state) of the spacecraft in real-time. Accurate state information is needed by the guidance system throughout all entry phases: pre-parachute deploy hypersonic entry, parachute deployment phase, and terminal guidance utilizing the main landing engines. The state information is also required for accurate determination of the landing site location on the Martian surface.

The determination of the current state of a spacecraft maneuvering in the Martian atmosphere is the subject of this paper. The task of the precision navigation system is to supply the current state of the spacecraft, as well as a measure of the uncertainty associated with the current state estimate, to the EDL systems. The proposed technique employs a recursive filter (that is, an extended Kalman filter, or EKF) processing data provided by a strapdown IMU and aided by external sensor data. The proposed external sensors are a scanning lidar and a phased-array radar, both of which supply altitude and velocity data, along with the hazard detection capability. The altitude and velocity data from both sensors are processed by the EKF.

The interaction of the aided-navigation system with the rest of the flight system is discussed. Of particular interest is the integrated guidance and navigation system performance. Results from simulated EDL scenarios representative of the 2007 launch opportunity will be presented. The entry scenario considered extends from cruise stage separation to touchdown.

EDL SCENARIO

Figure 1 shows the spacecraft sequence of events for the current Smart Lander baseline scenario. For a traditional direct-entry lander, the cruise stage separates from the entry vehicle minutes from atmospheric interface (defined as a radius of $3522.2km$ from the center of Mars). However, in the Smart Lander scenario, the cruise stage will perform a deflection maneuver after separation to provide communication and data relay for the entry body during EDL, a Smart Lander mission requirement. To support this requirement and to reduce the magnitude of the cruise stage deflection maneuver, cruise stage separation is 30 minutes before atmospheric entry. The entry body does not have external sensors or a direct-to-Earth radio link, so instead of initializing the onboard filter with a ground-based navigation solution for the entry interface conditions as for past direct-entry landers, a ground solution is generated for the separation time. The spacecraft state after separation is determined by propagation using the IMU data for the spacecraft dynamics with the ground-supplied separation state as initial conditions, a process known as dead reckoning.

After atmospheric entry, a closed-loop GN&C system is used to guide the vehicle through the hypersonic phase to the supersonic flight phases. The Smart Lander will have a lifting entry body (achieved via center of gravity offset or an aerodynamic trim tab), with the orientation of the lift vector controlled by the flight software to maintain a specified flight profile. This is a major change in Mars landers since Viking, which all have performed ballistic (zero-lift) entry, descent and landing, and is a major contributor to the reduction in landing errors.

Once specified criteria are achieved after the hypersonic flight phase, the supersonic parachute is deployed. Additional criteria are defined for the release of the supersonic parachute and the deployment of the subsonic parachute. Once the subsonic parachute deploy conditions are met, the heat shield and the backshell are released to allow the subsonic parachute to be deployed. Also at this time, the landing legs are deployed and the external sensors become operational.

The subsonic parachute deployment marks the beginning of the terminal EDL phase. From cruise stage separation to this point, all navigation output has been the result of dead reckoning, with no external data available to aid the IMU. From this point, the navigation filter is aided by altitude

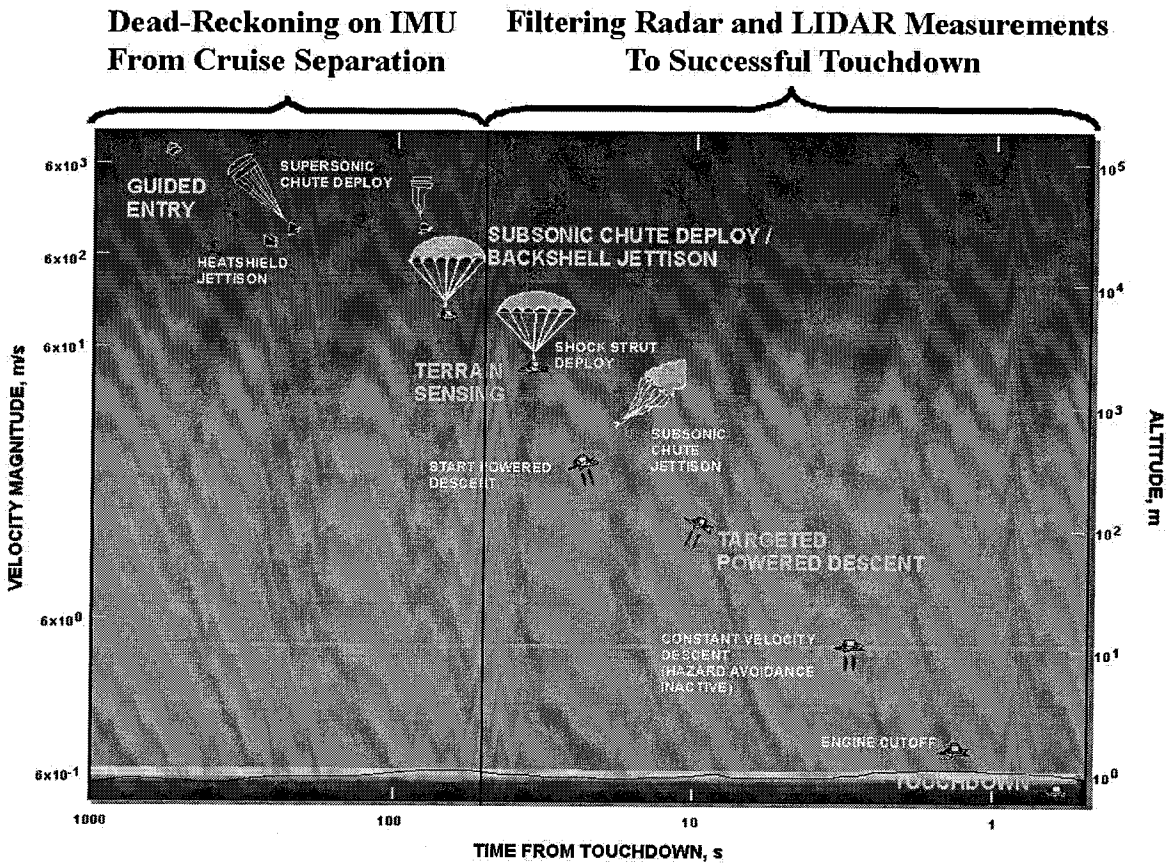

Figure 1 EDL Events

and velocity data from the radar. As shown in Figure 1, this occurs at approximately $8km$ altitude and $130sec$ from touchdown.

With the radar also providing terrain data, the hazard detection and avoidance functions begin, which require position and attitude data from the navigation filter. The scanning lidar will begin providing altitude and ground-relative velocity data at approximately $1500m$ altitude along with terrain data for hazard detection. These systems function through subsonic chute jettison and powered descent, stopping shortly before engine cutoff at $1m$ altitude. The sequence ends with a safe touchdown.

In terms of navigation functions for the EDL problem, there are two modes. The first is a dead-reckoning mode, which operates from cruise stage separation until the radar is operational, with the second from the start of radar data collection through landing. An additional mode before separation, which includes star camera data processing for attitude knowledge, may be added if the navigation filter starts operating before cruise stage separation (which is likely), but this mode will not be covered here. The major difference between these modes is the availability and type of external sensor data.

SENSOR DATA

Data used by the navigation filter for propagation and update are available from several sensors. The sensor suite for EDL includes an inertial measurement unit (IMU), a W-band phased-array

Table 1 IMU Specification

IMU	Power	Mass Volume	**IRU (1σ)** Bias Stability ARW Scale Factor	**ACC (1σ)** Bias Bias Noise Scale Factor	Comments
Litton LN100S	$30W$	$10.4\text{-}11.8Kg$ $28\times28\times17.8cm$ (IRU) $2.5(d)\times2.5cm$ (ACC)	$0.003\frac{deg}{hr}$ $0.0007\frac{deg}{\sqrt{hr}}$ $1ppm$	$0.025mg$ $5\mu g$ $100ppm$	Rad hard Delivery in 2001 No flight history
Honeywell MIMU	$34W$	$4.1kg$ $19.8(d)\times13.2cm$	$0.05\frac{deg}{hr}$ $0.01\frac{deg}{\sqrt{hr}}$ $5ppm$	$0.1mg$ $\sim 10\mu g$ $175ppm$	Rad hard $(>100Krads)$
Litton LN200	$12W$	$0.75Kg$ $8.9(d)\times8.9cm$	$3\frac{deg}{hr}$ $0.15\frac{deg}{\sqrt{hr}}$ $100ppm$	$1.5mg$ $35\mu g$ $1000ppm$	Flown on DS1 MER Baseline

radar and a scanning light detecting and ranging (lidar) sensor. The radar and lidar are designed to supply terrain data for the hazard detection and avoidance system, but the sensors will also supply data for updating the real-time position and attitude of the spacecraft during EDL. The emphasis of the sensor description below is on data that will be used for the latter function.

Inertial Measurement Unit (IMU)

An IMU includes an orthogonal accelerometer triad to measure velocity changes and an orthogonal gyroscope triad to record orientation changes. The data reported from the IMU are a velocity change vector and an attitude change vector over a specified accumulation time, referred to as the IMU rate. The internal sampling rate can be set to a few fixed values for a particular sensor, but the output velocity and attitude changes can be output from an IMU with greater flexibility. Specifications for the IMUs under consideration are shown in Table 1. These IMUs are all strapdown instruments, meaning they are rigidly attached to the spacecraft body and measure directly the velocity and attitude changes of the vehicle.

Phased-array Radar

A phased-array radar is under development that will provide altitude and velocity data for navigation. It can also be used as a supplement to the scanning lidar for hazard detection in a nominal scenario and can provide a backup capability to the scanning lidar. The radar is the primary sensor for navigation.

The proposed radar antenna location on the entry body means that the sensor is not exposed until the heat shield is released roughly $8km$ above the surface (see Figure 1), which limits the maximum operating range required for the sensor. Once the heat shield is released, the landing legs must be deployed and the sensor powered on before usable altitude or velocity data are available. The radar antenna is actually an array of small antennas each collecting range data simultaneously from different locations about the target to create a terrain map of the area.

Lidar

A lidar (light detecting and ranging) instrument is being developed as the primary sensor for hazard detection. The instrument sequentially collects range information in a grid around a defined target

and uses that frame of data to compute surface slopes and identify hazards around the target. Due to the sequential data collection, the data for the scan must be corrected for spacecraft motion during the scan, a major driver to the navigation state output rate. In contrast, the radar collects data at each point simultaneously and thus does not need motion compensation. This sensor begins collecting usable data at roughly $1500m$ altitude. The lidar will also provide altitude and velocity measurements relative to the target landing site.

For both the lidar and the radar, the method for computing altitude and surface-relative velocity for navigation from the collected data has not been finalized. For the current navigation filter, generic altitude and velocity measurement models will be used as placeholders for models that represent the final sensors with accuracies consistent with the design values. These generic models will be described later.

FILTER MODELS

Based on the EDL scenario described above, a set of filter models has been identified for development. Some of these models are dependent on sensors that are still under development and are more generic, while others are more mature. The set of models are described below, along with the set of filter parameters.

Filter State Vector

The state vector $\boldsymbol{X}$ includes all parameters that are solved for by the EKF and is written

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{r} & \boldsymbol{v} & \boldsymbol{\theta} & \boldsymbol{B}_\theta & \boldsymbol{B}_a & \boldsymbol{m} & B_h \end{bmatrix}^T$$

where the subvectors of $\boldsymbol{X}$ are defined in Table 2. The state vector will contain elements for estimated inertial position, inertial velocity and attitude. The remaining elements of $\boldsymbol{X}$ represent additional parameters that may be required based on sensor modeling assumptions and are not final.

Propagation Equations

Values for the state parameters above are computed at specified times based on initial values and mathematical models of the environment they represent. These propagation equations can be represented as first-order differential equations, as shown for the complete state vector:

$$\dot{\boldsymbol{X}} = \begin{bmatrix} \dot{\boldsymbol{r}} & \dot{\boldsymbol{v}} & \dot{\boldsymbol{\theta}} & \dot{\boldsymbol{B}}_\theta & \dot{\boldsymbol{B}}_a & \dot{\boldsymbol{m}} & \dot{B}_h \end{bmatrix}^T$$

For this implementation, most of the propagation relations are more naturally represented as discrete linear equations. The exceptions are the position and velocity elements, which are represented as first-order nonlinear differential equations since the gravitational acceleration (which cannot be measured by an accelerometer) must be computed. The complete set of propagation equations are described below.

Attitude propagation

The accelerometer and gyro data are used by the navigation filter to propagate the position, velocity and attitude in time. The approach for attitude propagation is discussed first, since the trajectory propagation depends on attitude.

The attitude is propagated as a scalar-last quaternion. The quaternion is based on the idea that an axis of rotation exists between two coordinate frames such that the rotation can be represented by

Table 2 State Vector

n	Symbol	Element	Units	Comments
1	$r(1)$	X Position	m	MCI frame, Cartesian
2	$r(2)$	Y Position	m	MCI frame, Cartesian
3	$r(3)$	Z Position	m	MCI frame, Cartesian
4	$v(1)$	X Velocity	m/s	MCI frame, Cartesian
5	$v(2)$	Y Velocity	m/s	MCI frame, Cartesian
6	$v(3)$	Z Velocity	m/s	MCI frame, Cartesian
7	$\theta(1)$	X Δ Attitude	rad	Body frame
8	$\theta(2)$	Y Δ Attitude	rad	Body frame
9	$\theta(3)$	Z Δ Attitude	rad	Body frame
10	$B_\theta(1)$	X Gyro Bias	rad	TBD Parameter
11	$B_\theta(2)$	Y Gyro Bias	rad	TBD Parameter
12	$B_\theta(3)$	Z Gyro Bias	rad	TBD Parameter
13	$B_a(1)$	X Accelerometer Bias	m/s^2	TBD Parameter
14	$B_a(2)$	Y Accelerometer Bias	m/s^2	TBD Parameter
15	$B_a(3)$	Z Accelerometer Bias	m/s^2	TBD Parameter
16	$m(1)$	Surface Slope 1	rad (?)	TBD Parameter
17	$m(2)$	Surface Slope 2	rad (?)	TBD Parameter
18	B_h	Altimeter Bias	m	TBD Parameter

a single angular rotation. A quaternion defines an axis of rotation (3 components) and a rotation angle about that vector (1 component). The inertial to body quaternion q_n is used here to replace the 3×3 inertial to body transformation matrix $\boldsymbol{C}_n$. The components of the quaternion are

$$q_n = \begin{bmatrix} q_{n_1} & q_{n_2} & q_{n_3} & q_{n_4} \end{bmatrix}^T$$

where q_{n_1}, q_{n_2} and q_{n_3} represent the vector direction and q_{n_4} represents the rotation angle.

The data from the IMU are an angular change in the axes of the spacecraft coordinate frame, denoted $\delta\boldsymbol{\theta}_n$ or the angular change in attitude measured in the body frame from t_{n-1} to t_n. With this information, the following quantities are defined:

$$\delta\boldsymbol{\theta}_n = \boldsymbol{\theta}_n - \boldsymbol{\theta}_{n-1}$$

$$\delta\boldsymbol{\theta}_n = \begin{bmatrix} \delta\theta_{n_x} & \delta\theta_{n_y} & \delta\theta_{n_z} \end{bmatrix}^T$$

$$\delta\boldsymbol{\theta}_{12} = \delta\boldsymbol{\theta}_1 + \delta\boldsymbol{\theta}_2$$

These definitions are used to describe the process.

The first step is to add the estimated gyro bias $\boldsymbol{B}_\theta$ to the raw angle data output:

$$\delta\boldsymbol{\theta}_n = \delta\boldsymbol{\theta}_{n_{raw}} + \boldsymbol{B}_\theta$$

The attitude data from the IMU is then converted to a quaternion δq_2, representing the angular

change $\delta\boldsymbol{\theta}_2$ in body coordinates, using the relation

$$\delta q_2 = \begin{bmatrix} \delta q_{2_1} \\ \delta q_{2_2} \\ \delta q_{2_3} \\ \delta q_{2_4} \end{bmatrix} = \begin{bmatrix} \frac{\delta\theta_{2_x}}{2} \\ \frac{\delta\theta_{2_y}}{2} \\ \frac{\delta\theta_{2_z}}{2} \\ 1 - \frac{\delta\theta_{12_x}^2 + \delta\theta_{12_y}^2 + \delta\theta_{12_z}^2}{32} \end{bmatrix}$$

Once the quaternion representing the attitude change is determined, the complete inertial to body quaternion at the current time t_2 is computed using the inertial to body quaternion from the previous time t_1:

$$q_2 = \delta q_2 q_1$$

where quaternion multiplication is defined as

$$q_1 q_2 = \begin{bmatrix} q_{1_4} q_{2_1} + q_{1_3} q_{2_2} - q_{1_2} q_{2_3} + q_{1_1} q_{2_4} \\ -q_{1_3} q_{2_1} + q_{1_4} q_{2_2} + q_{1_1} q_{2_3} + q_{1_2} q_{2_4} \\ q_{1_2} q_{2_1} - q_{1_1} q_{2_2} + q_{1_4} q_{2_3} + q_{1_3} q_{2_4} \\ -q_{1_1} q_{2_1} - q_{1_2} q_{2_2} - q_{1_3} q_{2_3} + q_{1_4} q_{2_4} \end{bmatrix}$$

An additional angular correction is applied to the quaternion after each filter update. This correction is due to the estimated attitude. This correction is applied to the attitude once per update. The relations used for this update are similar to those used for the propagation:

$$q_n^+ = \delta q_{\theta_{est}} q_n^-$$

$$\delta q_{\theta_{est}} = \begin{bmatrix} \frac{\theta_{x_{est}}}{2} \\ \frac{\theta_{y_{est}}}{2} \\ \frac{\theta_{z_{est}}}{2} \\ 1 - \frac{\theta_{x_{est}}^2 + \theta_{y_{est}}^2 + \theta_{z_{est}}^2}{32} \end{bmatrix}$$

In order to use the attitude quaternion for coordinate rotations from the body frame to the Mars centered inertial frame, the quaternion must be converted to a rotation matrix. The relation used to compute the rotation matrix $\boldsymbol{C}_n$ from a quaternion q_n is:

$$\boldsymbol{C}_n = \begin{bmatrix} q_{n_1}^2 - q_{n_2}^2 - q_{n_3}^2 + q_{n_4}^2 & 2q_{n_1}q_{n_2} + 2q_{n_3}q_{n_4} & 2q_{n_1}q_{n_3} - 2q_{n_2}q_{n_4} \\ 2q_{n_1}q_{n_2} - 2q_{n_3}q_{n_4} & -q_{n_1}^2 + q_{n_2}^2 - q_{n_3}^2 + q_{n_4}^2 & 2q_{n_2}q_{n_3} + 2q_{n_1}q_{n_4} \\ 2q_{n_1}q_{n_3} + 2q_{n_2}q_{n_4} & 2q_{n_2}q_{n_3} - 2q_{n_1}q_{n_4} & -q_{n_1}^2 - q_{n_2}^2 + q_{n_3}^2 + q_{n_4}^2 \end{bmatrix}$$

Trajectory propagation

The trajectory propagation is performed using a set of nonlinear differential equations, as described earlier. The differential equations for the trajectory calculation with time are:

$$\begin{aligned} \dot{\boldsymbol{r}}(t) &= \boldsymbol{v}(t) \\ \dot{\boldsymbol{v}}(t) &= \boldsymbol{C}_B^I(\boldsymbol{a}_{sf} + \boldsymbol{B}_a) + \boldsymbol{a}_g(\boldsymbol{r}) \end{aligned}$$

where $\boldsymbol{r}$ is the Cartesian position vector, $\boldsymbol{v}$ is the Cartesian velocity vector, $\boldsymbol{\omega}$ is the Cartesian angular rate, $\boldsymbol{C}_B^I$ is the body to inertial rotation matrix, $\boldsymbol{a}_{sf}$ is the measured acceleration in body coordinates, $\boldsymbol{B}_a$ is the computed accelerometer bias and $\boldsymbol{a}_g(\boldsymbol{r})$ is gravitational acceleration.

The propagation of the spacecraft position and velocity is done in a Mars centered inertial coordinate frame. Because of this, the velocity change from the IMU must be rotated from the body frame to this inertial frame. This transformation is computed using the attitude update relationships described in the previous section.

The accelerometer output is the accumulated acceleration over a specified time interval. The spacecraft velocity change reported by the IMU from time t_{n-1} to t_n is denoted $\delta\boldsymbol{v}_n$. It is important to note that the reported value does not include gravitational acceleration. The gravitational acceleration must be computed separately added to the velocity change from the IMU to get the correct total acceleration.

The rotation of $\delta\boldsymbol{v}_n$ to inertial coordinates is performed using the attitude at the midpoint of the interval t_{n-1} and t_n. This is approximated using the attitude quaternion at the beginning of the accumulation time q_{n-1} and performing an update with half the accumulated attitude change $\delta\boldsymbol{\theta}_n$, denoted $\delta\boldsymbol{\theta}_{mid_n}$:

$$\delta\boldsymbol{\theta}_{mid_n} = \frac{\delta\boldsymbol{\theta}_n}{2}$$

$$q_{mid_n} = \delta q_{mid_n} q_{n-1}$$

$$\delta\boldsymbol{v}_n^I = C_{mid_n}\delta\boldsymbol{v}_n$$

To compute the updated position and velocity at t_n, the value $\delta\boldsymbol{v}_{n-1}^I$ is also required. Due to the recursive nature of the propagation, this value is available from the previous update. For the current software, the velocity is converted to an acceleration, the specific force acceleration $\boldsymbol{a}_{sf_n}$:

$$\boldsymbol{a}_{sf_n} = \frac{\delta\boldsymbol{v}_n^I}{\Delta t}$$

$$\Delta t = t_n - t_{n-1}$$

To get the total acceleration, the gravitational acceleration must be computed and added to $\boldsymbol{a}_{sf_n}$ and $\boldsymbol{a}_{sf_{n-1}}$. For gravity terms up to J_2, the acceleration due to gravity can be computed using

$$\boldsymbol{a}_{g_n} = -\frac{\mu}{r_n^3}\begin{bmatrix} x_n\left(1 - J_2\frac{3}{2}\left(\frac{r_e}{r_n}\right)^2\left(5\sin^2 L_n - 1\right)\right) \\ y_n\left(1 - J_2\frac{3}{2}\left(\frac{r_e}{r_n}\right)^2\left(5\sin^2 L_n - 1\right)\right) \\ z_n\left(1 - J_2\frac{3}{2}\left(\frac{r_e}{r_n}\right)^2\left(5\sin^2 L_n - 3\right)\right) \end{bmatrix}$$

$$\boldsymbol{r}_n = \begin{bmatrix} x_n & y_n & z_n \end{bmatrix}^T \qquad r_n = \sqrt{x_n^2 + y_n^2 + z_n^2} \qquad \sin L_n = \frac{z_n}{r_n}$$

where μ is the gravitational parameter for Mars, J_2 is a gravity coefficient for Mars determined by observation, L_n is the geocentric longitude at time t_n and r_e is the equatorial radius of Mars. The position and velocity at t_{n-1} is required, but this information is available from inputs (if $t_{n-1} = t_0$) or from propagation over the previous time step. However, the position at t_n is the desired output from the current propagation, but is required to compute the gravitational acceleration needed for

the propagation. To compute $\boldsymbol{a}_{g_n}$, an approximate value for the position at t_n is determined using the position, velocity and total acceleration at t_{n-1}:

$$\boldsymbol{r}_{pred_n} = \boldsymbol{r}_{n-1} + \boldsymbol{v}_{n-1}\Delta t + \frac{1}{2}(a_{sf_{n-1}} + a_{g_{n-1}})\Delta t^2, \qquad \Delta t = (t_n - t_{n-1})$$

The total acceleration $\boldsymbol{a}_{tot}$ is thus:

$$\boldsymbol{a}_{tot_{n-1}} = \boldsymbol{a}_{sf_{n-1}} + \boldsymbol{a}_{g_{n-1}}$$

$$\boldsymbol{a}_{tot_n} = \boldsymbol{a}_{sf_n} + \boldsymbol{a}_{g_n}$$

$$\boldsymbol{a}_{g_n} = -\frac{\mu}{r^3_{pred_n}} \begin{bmatrix} x_{pred_n}\left(1 - J_2\frac{3}{2}\left(\frac{r_e}{r_{pred_n}}\right)^2 (5\sin^2 L_{pred_n} - 1)\right) \\ y_{pred_n}\left(1 - J_2\frac{3}{2}\left(\frac{r_e}{r_{pred_n}}\right)^2 (5\sin^2 L_{pred_n} - 1)\right) \\ z_{pred_n}\left(1 - J_2\frac{3}{2}\left(\frac{r_e}{r_{pred_n}}\right)^2 (5\sin^2 L_{pred_n} - 3)\right) \end{bmatrix}$$

With the total accelerations computed, the trajectory update is performed using a trapezoidal integration:

$$\boldsymbol{v}_n = \boldsymbol{v}_{n-1} + (\boldsymbol{a}_{tot_{n-1}} + \boldsymbol{a}_{tot_n})\frac{\Delta t}{2}$$

$$\boldsymbol{r}_n = \boldsymbol{r}_{n-1} + (\boldsymbol{v}_n + \boldsymbol{v}_{n-1})\frac{\Delta t}{2}$$

Bias Parameter Propagation

The remaining terms in $\boldsymbol{X}$ represent various sensor-related biases that are estimated in the Kalman filter. There are two types of noise models currently supported, namely a constant bias and a exponentially correlated random variable (ECRV) or first-order Markov process. The ECRV is a random variable whose autocorrelation function decreases exponentially with time. Depending on the error model chosen, the propagation model will change.

For a constant bias, the differential equation is simply

$$\dot{x} = 0$$

or, in discrete form,

$$x_{n+1} = x_n$$

This model is available for all bias modeling, but will mostly be used for the surface slope and altimeter bias errors. The only required input for this model (aside from error modeling, which is discussed later) is an initial value (usually set to zero).

For the ECRV model, the differential equation is

$$\dot{x} = -\frac{1}{\tau}x + w = -\beta x + w$$

where τ is the time constant used to define the decay rate and w is white noise. In discrete form,

$$x_{n+1} = e^{-\beta(t_{n+1}-t_n)}x_n + w_n$$

This model is also available for all bias modeling, but will primarily be used with the accelerometer and gyro bias models. The required input (apart from the error modeling) is the time constant τ.

Both error models, in discrete form, can be written as

$$x_{n+1} = \phi_{n,n+1} x_n + c_n w_n$$

$$\phi_{n,n+1} = \begin{cases} 1 & \text{for bias} \\ e^{-\beta(t_{n+1}-t_n)} & \text{for ECRV} \end{cases} \qquad c_n = \begin{cases} 0 & \text{for bias} \\ 1 & \text{for ECRV} \end{cases}$$

where $\phi_{n,n+1}$ is the state transition matrix (described below) for time propagation of x_n from t_n to t_{n+1}.

State Transition Matrix calculation

The state transition matrix (STM, $\boldsymbol{\Phi}$) is a linearized propagation matrix for the filter state vector. While not used to propagate the trajectory or the attitude, it is used to propagate the state error covariance from one filter update time to the next. It is also used to propagate the noise parameters between filter update times, as previously discussed.

The data required to compute the STM are the inertial position, velocity, acceleration and body to inertial rotation matrix at times t_{n-1} and t_n, along with $\delta\boldsymbol{\theta}_2$ and possible bias model values, all defined earlier.

Most of the STM is computed by integration of the partial derivatives of the propagation equations with respect to the estimated state vector, defined as $\boldsymbol{F}$. The integration performed, as before, is a trapezoidal integration represented as:

$$\boldsymbol{\Phi}_{n-1,n} = \boldsymbol{I} + (\boldsymbol{F}_{n-1} + \boldsymbol{F}_n)\frac{\Delta t}{2} + \boldsymbol{F}_n \boldsymbol{F}_{n-1}\frac{\Delta t^2}{2}$$

where $\boldsymbol{\Phi}_{n-1,n}$ is the STM from t_{n-1} to t_n, $\boldsymbol{I}$ is an identity matrix with the same dimension as $\boldsymbol{\Phi}_{n-1,n}$, $\boldsymbol{F}_n$ and $\boldsymbol{F}_{n-1}$ are $\boldsymbol{F}$ matrices evaluated at t_n and t_{n-1}, respectively and $\Delta t = t_n - t_{n-1}$.

Evaluation of $\boldsymbol{F}$ requires examination of the equations of motion for inertial navigation. The equations for trajectory and attitude calculation were presented earlier. The elements of $\boldsymbol{F}$ are the derivatives of $\dot{\boldsymbol{X}}$ with respect to the state vector $\boldsymbol{X}$ defined earlier. For the purposes of defining the elements of $\boldsymbol{F}$, it is convenient to define $\boldsymbol{F}$ in terms of submatrices, as shown:

$$\boldsymbol{F} = \frac{\partial \dot{\boldsymbol{X}}}{\partial \boldsymbol{X}} = \begin{bmatrix} \boldsymbol{F}^{3\times3}_{\dot{r}r} & \boldsymbol{F}^{3\times3}_{\dot{r}v} & \boldsymbol{F}^{3\times3}_{\dot{r}\theta} & \boldsymbol{F}^{3\times3}_{\dot{r}B_\theta} & \boldsymbol{F}^{3\times3}_{\dot{r}B_a} & \boldsymbol{F}^{3\times2}_{\dot{r}m} & \boldsymbol{F}^{3\times1}_{\dot{r}B_h} \\ \boldsymbol{F}^{3\times3}_{\dot{v}r} & \boldsymbol{F}^{3\times3}_{\dot{v}v} & \boldsymbol{F}^{3\times3}_{\dot{v}\theta} & \boldsymbol{F}^{3\times3}_{\dot{v}B_\theta} & \boldsymbol{F}^{3\times3}_{\dot{v}B_a} & \boldsymbol{F}^{3\times2}_{\dot{v}m} & \boldsymbol{F}^{3\times1}_{\dot{v}B_h} \\ \boldsymbol{F}^{3\times3}_{\dot{\theta}r} & \boldsymbol{F}^{3\times3}_{\dot{\theta}v} & \boldsymbol{F}^{3\times3}_{\dot{\theta}\theta} & \boldsymbol{F}^{3\times3}_{\dot{\theta}B_\theta} & \boldsymbol{F}^{3\times3}_{\dot{\theta}B_a} & \boldsymbol{F}^{3\times2}_{\dot{\theta}m} & \boldsymbol{F}^{3\times1}_{\dot{\theta}B_h} \\ \boldsymbol{F}^{3\times3}_{\dot{B}_\theta r} & \boldsymbol{F}^{3\times3}_{\dot{B}_\theta v} & \boldsymbol{F}^{3\times3}_{\dot{B}_\theta \theta} & \boldsymbol{F}^{3\times3}_{\dot{B}_\theta B_\theta} & \boldsymbol{F}^{3\times3}_{\dot{B}_\theta B_a} & \boldsymbol{F}^{3\times2}_{\dot{B}_\theta m} & \boldsymbol{F}^{3\times1}_{\dot{B}_\theta B_h} \\ \boldsymbol{F}^{3\times3}_{\dot{B}_a r} & \boldsymbol{F}^{3\times3}_{\dot{B}_a v} & \boldsymbol{F}^{3\times3}_{\dot{B}_a \theta} & \boldsymbol{F}^{3\times3}_{\dot{B}_a B_\theta} & \boldsymbol{F}^{3\times3}_{\dot{B}_a B_a} & \boldsymbol{F}^{3\times2}_{\dot{B}_a m} & \boldsymbol{F}^{3\times1}_{\dot{B}_a B_h} \\ \boldsymbol{F}^{2\times3}_{\dot{m}r} & \boldsymbol{F}^{2\times3}_{\dot{m}v} & \boldsymbol{F}^{2\times3}_{\dot{m}\theta} & \boldsymbol{F}^{2\times3}_{\dot{m}B_\theta} & \boldsymbol{F}^{2\times3}_{\dot{m}B_a} & \boldsymbol{F}^{2\times2}_{\dot{m}m} & \boldsymbol{F}^{2\times1}_{\dot{m}B_h} \\ \boldsymbol{F}^{1\times3}_{\dot{B}_h r} & \boldsymbol{F}^{1\times3}_{\dot{B}_h v} & \boldsymbol{F}^{1\times3}_{\dot{B}_h \theta} & \boldsymbol{F}^{1\times3}_{\dot{B}_h B_\theta} & \boldsymbol{F}^{1\times3}_{\dot{B}_h B_a} & \boldsymbol{F}^{1\times2}_{\dot{B}_h m} & \boldsymbol{F}^{1\times1}_{\dot{B}_h B_h} \end{bmatrix}$$

where for each $\boldsymbol{F}^{n\times m}_{\dot{a}b}$, $\dot{a}$ is the set of n differential equations whose partials with respect to each of m elements of b are computed, with the resulting submatrix of dimension $n \times m$. Recall that each row of $\boldsymbol{F}$ corresponds to a propagation equation, and each column of $\boldsymbol{F}$ corresponds to an element of the state vector defined earlier.

The values of $\boldsymbol{F}$ corresponding to the $\dot{\boldsymbol{r}}(t)$ equation are all zero, except for the partial with respect to the spacecraft velocity:

$$\boldsymbol{F}_{\dot{r}v} = \boldsymbol{I}_{3\times3}$$

The partials of $\dot{\boldsymbol{v}}(t)$ with respect to $\boldsymbol{X}$ are more complicated. Using the differential equation for $\dot{\boldsymbol{v}}(t)$ and computing variations,

$$\delta\dot{\boldsymbol{v}}(t) = \boldsymbol{C}_B^I \delta\boldsymbol{a}_{sf}^B + (\delta\boldsymbol{C}_B^I)\boldsymbol{C}_B^I\boldsymbol{a}_{sf}^B + \delta\boldsymbol{a}_g^B(\boldsymbol{r})$$

The only term dependent on $\boldsymbol{r}$ is $\delta\boldsymbol{a}_g^B(\boldsymbol{r})$. Using the point-mass only definition of gravitational acceleration, the partials are:

$$\boldsymbol{F}_{\dot{v}r} = -\frac{\mu}{r^3}\begin{bmatrix} 1-\frac{3x^2}{r^2} & -\frac{3xy}{r^2} & -\frac{3xz}{r^2} \\ -\frac{3yx}{r^2} & 1-\frac{3y^2}{r^2} & -\frac{3yz}{r^2} \\ -\frac{3zx}{r^2} & -\frac{3zy}{r^2} & 1-\frac{3z^2}{r^2} \end{bmatrix}$$

where μ is the gravitational parameter for Mars. The partial with respect to $\boldsymbol{v}$ is zero:

$$\boldsymbol{F}_{\dot{v}r} = \boldsymbol{0}_{3\times3}$$

The partial with respect to $\boldsymbol{\theta}$ depends only on the small variations in the attitude, or $(\delta\boldsymbol{C}_B^I)\boldsymbol{C}_B^I\boldsymbol{a}_{sf}^B$. The term $\delta\boldsymbol{C}_B^I$ can be obtained by writing $\boldsymbol{C}_B^I$ in terms of Euler angles and making small angle approximations. With all three terms expanded:

$$\begin{aligned}
(\delta\boldsymbol{C}_B^I)\boldsymbol{C}_B^I\boldsymbol{a}_{sf}^B &= \begin{bmatrix} 1 & \theta(3) & -\theta(2) \\ -\theta(3) & 1 & \theta(1) \\ \theta(2) & -\theta(1) & 1 \end{bmatrix}\begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix}\begin{bmatrix} a_{sf}^B(1) \\ a_{sf}^B(2) \\ a_{sf}^B(3) \end{bmatrix} \\
&= \begin{bmatrix} 1 & \theta(3) & -\theta(2) \\ -\theta(3) & 1 & \theta(1) \\ \theta(2) & -\theta(1) & 1 \end{bmatrix}\begin{bmatrix} a_{sf}^I(1) \\ a_{sf}^I(2) \\ a_{sf}^I(3) \end{bmatrix} \\
&= \begin{bmatrix} a_{sf}^I(1) + \theta(3)a_{sf}^I(2) - \theta(2)a_{sf}^I(3) \\ -\theta(3)a_{sf}^I(1) + a_{sf}^I(2) + \theta(1)a_{sf}^I(3) \\ \theta(2)a_{sf}^I(1) - \theta(1)a_{sf}^I(2) + a_{sf}^I(3) \end{bmatrix}
\end{aligned}$$

Taking the partial with respect to $\boldsymbol{\theta}$ gives the resulting submatrix:

$$\boldsymbol{F}_{\dot{v}\theta} = \begin{bmatrix} 0 & -a_{sf}^I(3) & a_{sf}^I(2) \\ a_{sf}^I(3) & 0 & -a_{sf}^I(1) \\ -a_{sf}^I(2) & a_{sf}^I(1) & 0 \end{bmatrix}$$

The partial with respect to the gyro bias $\boldsymbol{B}_\theta$ is zero:

$$\boldsymbol{F}_{\dot{v}B_\theta} = \boldsymbol{0}_{3\times3}$$

The partial with respect to the accelerometer bias is changed by variations in the acceleration, or $\boldsymbol{C}_B^I\delta\boldsymbol{a}_{sf}^B$. The resulting matrix is:

$$\boldsymbol{F}_{\dot{v}B_a} = \boldsymbol{C}_B^I$$

The remaining velocity differential equation terms are zero.

The remaining differential equations are for parameters that are modeled as biases or ECRVs. For these terms, the partial derivatives are not a function of other elements in the state vector, so the only nonzero terms will be along the diagonal, or $\boldsymbol{F}_{\dot{B}_\theta B_\theta}$, $\boldsymbol{F}_{\dot{B}_a B_a}$, $\boldsymbol{F}_{\dot{m}m}$ and $\boldsymbol{F}_{\dot{B}_h B_h}$. The diagonal value of each element in these submatrices will by 0 for the bias parameters and $-\frac{1}{\tau}$ for the ECRVs.

Covariance Propagation

With the state transition matrix computed as above, the necessary pieces to propagate the state covariance matrix in time are available. The relationship used to propagate the covariance from t_{i-1} to t_i is

$$\boldsymbol{P}_i^{(-)} = \boldsymbol{\Phi}_{i-1,i}\boldsymbol{P}_{i-1}^{(+)}\boldsymbol{\Phi}_{i-1,i}^T + \boldsymbol{Q}_{i-1}$$

where $\boldsymbol{P}_{()}^{(-)}$ refers to the covariance matrix before measurement update, $\boldsymbol{P}_{()}^{(+)}$ refers to the covariance matrix after measurement update, $\boldsymbol{\Phi}$ is the state transition matrix defined above and $\boldsymbol{Q}$ is the process noise covariance.

Due to the long dead reckoning propagation required for this application, additional fidelity in the process noise is desired. Instead of direct numerical input of the parameters in $\boldsymbol{Q}$, relationships between the accelerometer and gyroscope errors specified in Table 1 and the state vector terms were developed. This formulation will be reported in a future paper. This allows for sensitivity analysis based on the IMU specifications shown in Table 1 for the instruments under consideration and a more accurate representation of the dead-reckoning error growth.

Generic Sensor Models

Since the filter under development is intended for use with the Mars Smart Lander, sensor models are needed that accurately represent the actual sensors. Since the sensors are still under development, generic altimeter and velocity measurement models have been developed for analysis purposes.

For the altimeter, the model is a scalar distance from the spacecraft to the proposed landing site. The measurement model used is

$$h_{h_{obs}} = \sqrt{r(1)^2 + r(2)^2 + r(3)^2} - r_{planet} + B_h$$

where $r()$ are the spacecraft Mars-centered inertial position components, r_{planet} is the distance from the center of Mars to the landing site and B_h is a bias on the altimeter measurement. The measurement partials, based on the state vector defined earlier, are

$$\boldsymbol{H}_h = \frac{\partial \boldsymbol{h}_h}{\partial \boldsymbol{X}} = \begin{bmatrix} \frac{r(1)}{|\boldsymbol{r}|} & \frac{r(2)}{|\boldsymbol{r}|} & \frac{r(3)}{|\boldsymbol{r}|} & 0 & 0 & 0 & \cdots & 1 \end{bmatrix}$$

The content of the velocity measurements returned by both sensors is currently under development. Each sensor will determine the vertical velocity along with velocity normal to the beam direction. As a first cut at a model, it is assumed that a vector velocity relative to the landing site in inertial coordinates is returned, or

$$\boldsymbol{h}_{\boldsymbol{V}_{obs}} = \boldsymbol{v}_{sc} - \boldsymbol{v}_{ls} = \begin{bmatrix} X(4) - v_{ls}(1) \\ X(5) - v_{ls}(2) \\ X(6) - v_{ls}(3) \end{bmatrix}$$

where $\boldsymbol{v}_{sc}$ is the inertial velocity of the spacecraft and $\boldsymbol{v}_{ls}$ is the inertial velocity of the landing site. For the Mars-centered inertial frame assumed here,

$$\boldsymbol{v}_{ls} = \boldsymbol{\omega}_m \times \boldsymbol{r}_{ls} \qquad\qquad \boldsymbol{\omega}_m = \begin{bmatrix} 0 & 0 & \omega_m \end{bmatrix}^T$$

where $\boldsymbol{r}_{ls}$ is the inertial position of the landing site and ω_m is the rotation rate of Mars. The inertial frame assumed here is the same as that for the estimated state, so the inertial velocity of

the spacecraft above will be the velocity in the estimated state, as defined in Table 2. The partials for this measurement are

$$\boldsymbol{H_V} = \frac{\partial \boldsymbol{h_V}}{\partial \boldsymbol{X}} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & \cdots & 0 \end{bmatrix}$$

Update Equations

Referring again to Figure 1, there is a period after heat shield release and before touchdown where sensor data are collected using the sensors described above. Information from these sensors is used to update the state vector. The final formulation of the update equations used is still in progress, but a standard update formulation has been implemented for testing. The relations used for the update in the current formulation are

$$\begin{aligned} \boldsymbol{K}_i &= \boldsymbol{P}_i \boldsymbol{H}_i^T (\boldsymbol{H}_i \boldsymbol{P}_i^{(-)} \boldsymbol{H}_i^T + \boldsymbol{R}_i)^{-1} \\ \boldsymbol{P}_i^{(+)} &= (\boldsymbol{I} - \boldsymbol{K}_i \boldsymbol{H}_i) \boldsymbol{P}_i^{(-)} (\boldsymbol{I} - \boldsymbol{K}_i \boldsymbol{H}_i)^T + \boldsymbol{K}_i \boldsymbol{R}_i \boldsymbol{K}_i^T \\ \boldsymbol{X}_i^{(+)} &= \boldsymbol{X}_i^{(-)} + \boldsymbol{K}_i (\boldsymbol{y}_i - \boldsymbol{h}_i(\boldsymbol{X}_i^{(-)})) \end{aligned}$$

where $\boldsymbol{K}$ is the Kalman gain used for the update, $\boldsymbol{y}$ is the observation vector and $\boldsymbol{y}_i - \boldsymbol{h}_i(\boldsymbol{X}_i^{(-)})$ represents the measurement residual at time i.

FILTER ARCHITECTURE

The navigation filter under development for Mars Smart Lander has been targeted for several different simulation environments, which requires an architecture that allows the filter to be installed in each tool with minimal interface modification. The simulation environments currently targeted include a terminal GN&C simulation and a real-time testbed under development at the Jet Propulsion Laboratory, the main Mars Smart Lander high-fidelity EDL simulation environment under development at the Langley Research Center, and the guidance design environment at the Johnson Space Center. The general idea is to develop the navigation filter as a module with a single function entry point, with all data required for processing passed through a well-defined function interface.

A block diagram of the proposed terminal GN&C system is shown in Figure 2. The ovals on the right side of the figure represent environment models of interest. Interaction between the environment and the terminal GN&C system is through the thrusters, lidar, radar and IMU boxes, which represent simulation models of hardware systems. The remaining boxes to the left of the hardware models represent GN&C software under development. The only information that the onboard simulation gains from the environment is through the hardware sensors, which is important to insure separation between the environment models and the spacecraft model. The tight interaction of navigation (labeled Position and Attitude Estimator here) with the guidance and control functions is apparent. The diagram for earlier phases of EDL will be similar, but will not include the lidar and radar blocks since the sensors are not yet operational. Hazard detection and avoidance functions will also be idle before the lidar and radar sensors become active.

A more detailed diagram of the navigation filter and sensors supplying data to the navigation filter is found in Figure 3. Data processing is initiated by a main executive routine that calls the navigation task when required. The standard processing procedure is that each sensor will have one processing pass for each call to the filter, but the filter routines allow for multiple available data points for each sensor. The interface between the navigation filter and the various sensors is a data

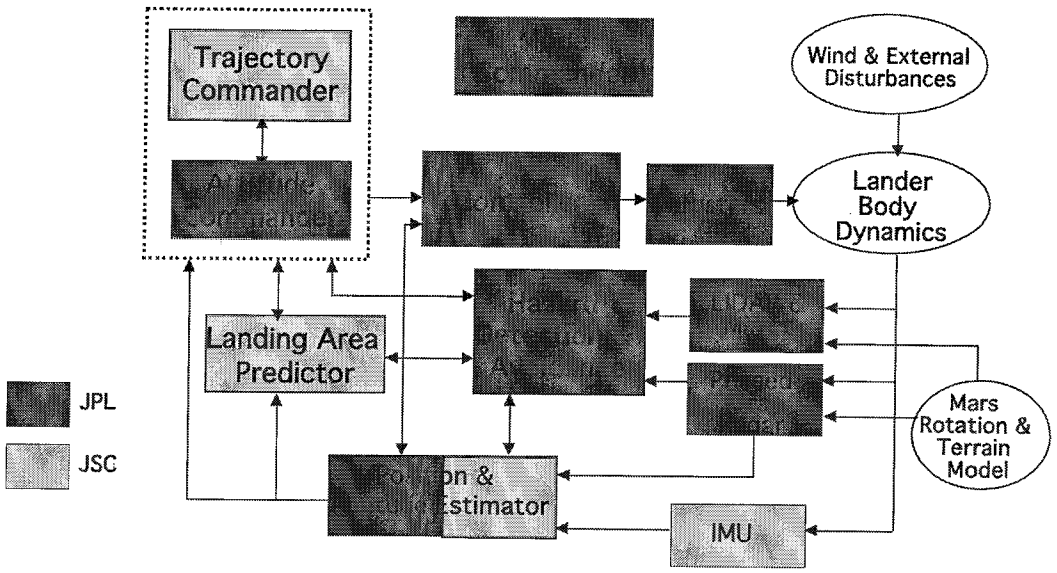

Figure 2 Terminal GN&C Block Diagram

query initiated by the navigation filter routine via sequential function calls for each instrument. If IMU data are available, the trajectory and the attitude are propagated in time based on the latest available state vector and the IMU data. If radar or lidar data are also available, filter updates of the state and covariance are performed using the data from each sensor. Checks are made as well to determine if altitude measurements, velocity measurements or both are present and processing occurs accordingly. If radar and lidar data are available for state update and no IMU data is available for a state propagation, the data are stored until propagation data from the IMU are available. If none of the sensors pass data to the navigation filter, no processing is done. Measurement processing includes propagation and update of the covariance matrix and updates to the trajectory, attitude and bias parameters.

RESULTS

The filter development effort is at the point where simulation of the dead-reckoning phase is possible, or from cruise stage separations to subsonic parachute deployment. Component testing of the modules in the filter is under way, along with more complete simulations to support the Mars Smart Lander design process. An example of this support is a sensitivity analysis to evaluate the three IMUs whose specifications were shown in Table 1. The approach used for this evaluation is a Monte Carlo analysis using a simulation at the Johnson Space Center, which is a combined guidance/navigation testbed. Cases were run from cruise stage separation through the dead reckoning phase with the parameter list shown in Table 3. These entries also represent the parameters that can be added or removed from each run to determine the impact of specific parameters on the guidance and navigation performance.

In addition to the enhanced IMU error modeling described earlier, there are techniques available to improve the performance of dead-reckoning propagation. The majority of the error is due to uncertainty in the calibration of the accelerometers and gyros, so two techniques are used to reduce the impact of this error. The first is to reject IMU output when the sensed attitude or velocity change is below a certain value, usually specified as a percentage of the nominal bias. This process, known as thresholding, allows the removal of IMU output to the integrator that is due to noise

Table 3 Sensitivity Analysis Parameters

Number	Component	Number	Component
1	X Nav	29	Gyro Misalign 5
2	Y Nav	30	Gyro Misalign 6
3	Z Nav	31	Rotation 1 Nav
4	Vx Nav	32	Rotation 2 Nav
5	Vy Nav	33	Rotation 3 Nav
6	Vz Nav	34	X Env
7	Accel X Case Bias	35	Y Env
8	Accel Y Case Bias	36	Z Env
9	Accel Z Case Bias	37	Vx Env
10	Accel Misalign 1	38	Vy Env
11	Accel Misalign 2	39	Vz Env
12	Accel Misalign 3	40	Rotation 1 Env
13	Accel Misalign 4	41	Rotation 2 Env
14	Accel Misalign 5	42	Rotation 3 Env
15	Accel Misalign 6	43	CG X Env
16	Accel X Case SF	44	CG Y Env
17	Accel Y Case SF	45	CG Z Env
18	Accel Z Case SF	46	CG X Nav
19	Gyro X Case Bias	47	CG Y Nav
20	Gyro Y Case Bias	48	CG Z Nav
21	Gyro Z Case Bias	49	Aero
22	Gyro X Case SF	50	Accel X Case Noise
23	Gyro Y Case SF	51	Accel Y Case Noise
24	Gyro Z Case SF	52	Accel Z Case Noise
25	Gyro Misalign 1	53	Gyro X Case Noise
26	Gyro Misalign 2	54	Gyro Y Case Noise
27	Gyro Misalign 3	55	Gyro Z Case Noise
28	Gyro Misalign 4		

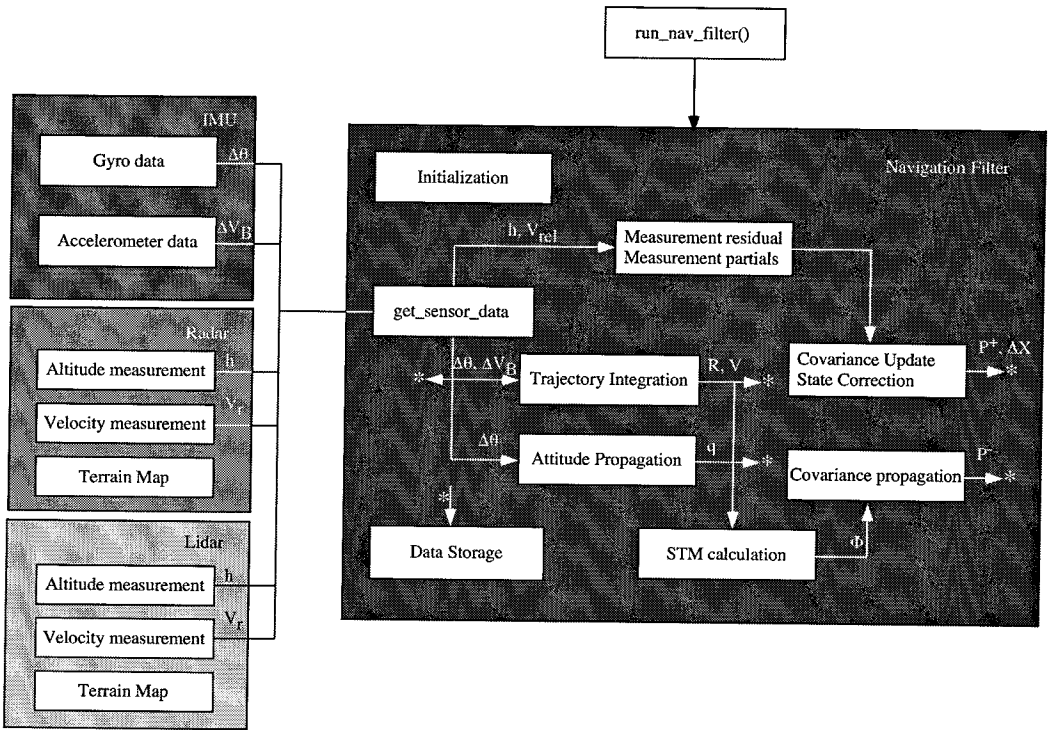

Figure 3 Navigation Filter Data Flow

only. Care is needed to select values consistent with the problem being solved to remove only data that is noise only but retain sensitivity to small signals. For this case, the IMU output will be small between separation and atmospheric interface since the only activity is thruster-based attitude maintenance to within a specified attitude deadband. The thresholds must be set low enough to sense the atmosphere as early as possible, but high enough to eliminate noise-only IMU output. The values to be used are still under investigation. The second error reduction technique is to assume calibration of the IMU biases before separation. This places some assumptions on when the IMU is turned on relative to separation and the availability of external data before separation. The approach assumed here is that the IMU is turned on at the latest before the final midcourse maneuver for accelerometer calibration and that star camera data is available for calibration of the gyro biases at a minimum from the final midcourse maneuver to separation. Although calibration is likely to be in the baseline plan, analysis of performance without calibration is needed. In the same way, thresholding will most likely be included in the flight code, but analysis with and without it will be done.

Sensitivity analysis has been performed for two types of results, referred to as navigation results and delivery results. Navigation results are those for the filter alone, or assuming perfect guidance, with comparison between the navigation and the environment truth states. Delivery results are those with guidance included, with comparisons between the navigation and the guidance truth states. Results for propagation from separation to the end of the dead-reckoning phase with the LN-100s IMU are tabulated in Table 4 for the major contributors to the position and velocity errors. These results assume that thresholding and calibration of the biases has been done. Comparisons of the case with all errors active are made as percentage reduction in the overall position and velocity errors when the specified parameter is assumed to contribute no error. For reference, the RSS position error with all parameters active is $1,500m$ and the RSS velocity error is $4\frac{m}{s}$. In terms

Table 4 Change from full case for each parameter set using the LN-100s

Input Deck	Error Component Removed	% Drop in Position Error	% Drop in Velocity Error
2	Initial NAV pos. & vel.	39.7%	3.5%
9	NAV initial attitude	3.1%	23.1%
11	Delivery attitude	3.4%	24.2%
17	All IMU errors	9.8%	1.0%

of position error, the major contributors are the errors in the initial position and velocity before separation, with the aggregate IMU error second. For velocity error, the major contribution is from attitude error.

Analysis using the Honeywell MIMU has also been performed, but the results are not presented here. The general trends seen with the LN-100s results above follow, but the IMU error contribution is larger, as is the overall error at the end of the simulation. As a percentage, the IMU error and initial navigation position and velocity error contributions are closer in magnitude than for the LN-100s.

FUTURE WORK

The sensitivity analysis above is just the beginning of the analysis required to understand the navigation problem for EDL. In addition to the sensitivity analysis described above for the dead reckoning phase, additional performance testing is required. For example, this analysis does not include the other elements of the onboard algorithms, namely guidance and control, which will contribute additional error to the overall knowledge of the spacecraft state.

In addition, the processing of external measurements has been developed but not tested in an integrated sense. This addition will allow for end-to-end simulation of EDL. Sensor models that accurately represent the proposed systems are under development along with the hardware. These issues will be addressed with the development of an end-to-end simulation capability, culminating in a hardware-in-the-loop testbed running candidate flight software.

ACKNOWLEDGEMENTS

The authors would like to thank Sam Thurman, Jeff Umland and Ed Wong from the Mars Smart Lander EDL team for their support of this effort. This development effort is funded by the Mars Technology Program, managed by Samad Hayati and the Mars Smart Lander Pre-Project, managed by Mike Sander.

This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, the Johnson Space Center and The University of Texas at Austin, under a contract with the National Aeronautics and Space Administration.

REFERENCES

1. Bate, R. R., Mueller, D. D. and White, J. E., *Fundamentals of Astrodynamics*, Dover Publications Inc., New York, 1971.

2. Britting, K., *Inertial Navigation System Analysis*, Wiley Interscience, New York, 1971.

3. Brown, R. G. and Hwang, P. Y. C., *Introduction to Random Signals and Applied Kalman Filtering*, 2^{nd} Edition, J. Wiley, 1992.

4. Gelb, A, *Applied Optimal Estimation*, MIT Press, 1992.

5. Savage, P. G. , "Strapdown Inertial Navigation Lecture Notes," Strapdown Associates, Inc., Maple Plain, MN, 55359, July 1997 (sixth printing).

6. Wertz J. R. (Ed.), *Spacecraft Attitude Determination and Control*, D. Reidel Publishing Co., Boston, 1978.